

**LA-UR-21-28980**

Approved for public release; distribution is unlimited.

**Title:** Fully Implicit Time Integration: Fast Solvers and Implicit-Explicit Schemes

**Author(s):** Southworth, Benjamin Scott  
Buvoli, Tomasso  
Krzysik, Oliver  
Pazner, Will  
De Sterck, Hans

**Intended for:** Seminar

**Issued:** 2021-09-13

---

**Disclaimer:**

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# Fully Implicit Time Integration: Fast Solvers and Implicit-Explicit Schemes

Ben Southworth, Oliver Krzysik,  
Tommaso Buvoli, Will Pazner,  
Hans De Sterck,

14 Sep. 2021



Managed by Triad National Security, LLC for the U.S. Department of Energy's NNSA

# Implicit integration

Important for when the explicit time step constraint (physical or numerical) is <sup>tiny</sup>.

# Implicit integration

Important for when the explicit time step constraint (physical or numerical) is <sup>tiny</sup>.

## Options:

- Diagonally implicit Runge-Kutta ([Easy to implement/solve](#))

# Implicit integration

Important for when the explicit time step constraint (physical or numerical) is <sup>tiny</sup>.

## Options:

- Diagonally implicit Runge-Kutta (Limited to stage-order 1)

# Implicit integration

Important for when the explicit time step constraint (physical or numerical) is <sup>tiny</sup>.

## Options:

- Diagonally implicit Runge-Kutta (**Limited to stage-order 1**)
- BDF methods (**Accurate on stiff nonlinear PDEs**)

# Implicit integration

Important for when the explicit time step constraint (physical or numerical) is  $\text{tiny}$ .

## Options:

- Diagonally implicit Runge-Kutta (Limited to stage-order 1)
- BDF methods (Unstable for advective problems and order  $\gtrapprox 2$ )

# Implicit integration

Important for when the explicit time step constraint (physical or numerical) is <sup>tiny</sup>.

## Options:

- Diagonally implicit Runge-Kutta (**Limited to stage-order 1**)
- BDF methods (**Unstable for advective problems and order  $\gtrless 2$** )
- Fully implicit Runge-Kutta (**(Very) high-order accuracy, good stability**)

# Implicit integration

Important for when the explicit time step constraint (physical or numerical) is tiny.

Options:

- Diagonally implicit Runge-Kutta (Limited to stage-order 1)
- BDF methods (Unstable for advective problems and order  $\gtrapprox 2$ )
- Fully implicit Runge-Kutta (Difficult/expensive to solve)

$$\left( \begin{bmatrix} M & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & M \end{bmatrix} - \delta t \begin{bmatrix} a_{11}\mathcal{L}_1 & \dots & a_{1s}\mathcal{L}_1 \\ \vdots & \ddots & \vdots \\ a_{s1}\mathcal{L}_s & \dots & a_{ss}\mathcal{L}_s \end{bmatrix} \right) \begin{bmatrix} \mathbf{k}_1 \\ \vdots \\ \mathbf{k}_s \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_s \end{bmatrix}.$$

## DG in time

DG in time discretizations of linear ODEs yield linear systems:

$$\left( \begin{bmatrix} \delta_{11}M & \delta_{1s}M \\ \ddots & \ddots \\ \delta_{s1}M & \delta_{ss}M \end{bmatrix} - \delta t \begin{bmatrix} m_{11}\mathcal{L}_1 & \dots & m_{1s}\mathcal{L}_1 \\ \vdots & \ddots & \vdots \\ m_{s1}\mathcal{L}_s & \dots & m_{ss}\mathcal{L}_s \end{bmatrix} \right) \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_s \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_s \end{bmatrix}$$

- $\{m_{ij}\} \sim$  temporal mass matrix,  $\{\delta_{ij}\} \sim$  DG weak derivative with upwind numerical flux.
- Degree- $p$  DG using  $(p+1)$ -point Radau quadrature equivalent to Radau IIA collocation.

# Table of Contents

Fast solvers for fully implicit integration

Numerical results

Fully implicit-explicit integration

Polynomial integrators

Implicit-explicit Radau

Numerical results

# Key assumptions

Consider Butcher tableaux  $\{A_0, \mathbf{b}_0\}$  and system of ODEs:

$$M\mathbf{u}'(t) = \mathcal{N}(\mathbf{u}, t) \quad \text{in } (0, T], \quad \mathbf{u}(0) = \mathbf{u}_0.$$

# Key assumptions

Consider Butcher tableaux  $\{A_0, \mathbf{b}_0\}$  and system of ODEs:

$$M\mathbf{u}'(t) = \mathcal{N}(\mathbf{u}, t) \quad \text{in } (0, T], \quad \mathbf{u}(0) = \mathbf{u}_0.$$

## Assumption 1.

Assume eigenvalues of  $A_0$  have positive real part.

## Assumption 2.

Let  $\mathcal{L}$  be the linear(ized) spatial operator; assume field-of-values (FOV)  $W(\mathcal{L}) = \{\langle W\mathbf{x}, \mathbf{x} \rangle / \|\mathbf{x}\|^2 : \mathbf{x} \neq \mathbf{0}\} \leq 0$ .

- Note,  $\|e^{t\mathcal{L}}\| \leq 1$  for all  $t \geq 0$  i.f.f.  $W(\mathcal{L}) \leq 0$ .

# Approximate Schur decomposition

Linearized systems take the form:

$$\left( \begin{bmatrix} I & \mathbf{0} \\ \ddots & I \\ \mathbf{0} & I \end{bmatrix} - \delta t \begin{bmatrix} a_{11}\mathcal{L}_1 & \dots & a_{1s}\mathcal{L}_1 \\ \vdots & \ddots & \vdots \\ a_{s1}\mathcal{L}_s & \dots & a_{ss}\mathcal{L}_s \end{bmatrix} \right) \begin{bmatrix} \mathbf{k}_1 \\ \vdots \\ \mathbf{k}_s \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_s \end{bmatrix}.$$

# Approximate Schur decomposition

Linearized systems take the form:

$$\left( A_0^{-1} \otimes I - \delta t \begin{bmatrix} \mathcal{L}_1 & & \\ & \ddots & \\ & & \mathcal{L}_s \end{bmatrix} \right) (A_0 \otimes I) \begin{bmatrix} \mathbf{k}_1 \\ \vdots \\ \mathbf{k}_s \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_s \end{bmatrix}.$$

# Approximate Schur decomposition

Linearized systems take the form:

$$\left( A_0^{-1} \otimes I - \delta t \begin{bmatrix} \mathcal{L}_1 & & \\ & \ddots & \\ & & \mathcal{L}_s \end{bmatrix} \right) (A_0 \otimes I) \begin{bmatrix} \mathbf{k}_1 \\ \vdots \\ \mathbf{k}_s \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_s \end{bmatrix}.$$

Let  $A_0^{-1} = Q_0 R_0 Q_0^T$  (real Schur decomposition),  $\hat{\mathbf{f}} := (Q_0^T \otimes I)\mathbf{f}$ , and  $\hat{\mathbf{k}} := (R_0^{-1} Q_0^T \otimes I)\mathbf{k}$ . Equivalent:

$$\left( R_0 \otimes I - (Q_0^T \otimes I) \begin{bmatrix} \mathcal{L}_1 & & \\ & \ddots & \\ & & \mathcal{L}_s \end{bmatrix} (Q_0 \otimes I) \right) \hat{\mathbf{k}} = \hat{\mathbf{f}}.$$

# Approximate Schur decomposition

Linearized systems take the form:

$$\left( A_0^{-1} \otimes I - \delta t \begin{bmatrix} \mathcal{L}_1 & & \\ & \ddots & \\ & & \mathcal{L}_s \end{bmatrix} \right) (A_0 \otimes I) \begin{bmatrix} \mathbf{k}_1 \\ \vdots \\ \mathbf{k}_s \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_s \end{bmatrix}.$$

Let  $A_0^{-1} = Q_0 R_0 Q_0^T$  (real Schur decomposition),  $\hat{\mathbf{f}} := (Q_0^T \otimes I)\mathbf{f}$ , and  $\hat{\mathbf{k}} := (R_0^{-1} Q_0^T \otimes I)\mathbf{k}$ . Equivalent:

$$\left( R_0 \otimes I - (Q_0^T \otimes I) \begin{bmatrix} \mathcal{L}_1 & & \\ & \ddots & \\ & & \mathcal{L}_s \end{bmatrix} (Q_0 \otimes I) \right) \hat{\mathbf{k}} = \hat{\mathbf{f}}.$$

$Q_0^T Q_0 = I \implies$  If  $\{\mathcal{L}_j\}$  commute, right term is block-diagonal.

# Linear block preconditioning

Requires solution of  $s/2$  systems

$$\begin{bmatrix} \eta I - \delta t \mathcal{L}_1 & \phi I \\ -\frac{\beta^2}{\phi} I & \eta I - \delta t \mathcal{L}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{k}}_1 \\ \hat{\mathbf{k}}_2 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{f}}_1 \\ \hat{\mathbf{f}}_2 \end{bmatrix} \quad (\eta > 0).$$

Convergence of block-prec Krylov defined by preconditioning of Schur complement,

$$S := \eta I - \delta t \mathcal{L}_2 + \beta^2 (\eta I - \delta t \mathcal{L}_1)^{-1}$$

# Linear block preconditioning

Requires solution of  $s/2$  systems

$$\begin{bmatrix} \eta I - \delta t \mathcal{L}_1 & \mathbf{0} \\ -\frac{\beta^2}{\phi} I & \frac{\eta^2 + \beta^2}{\eta} I - \delta t \mathcal{L}_2 \end{bmatrix}^{-1} \left( \begin{bmatrix} \eta I - \delta t \mathcal{L}_1 & \phi I \\ -\frac{\beta^2}{\phi} I & \eta I - \delta t \mathcal{L}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{k}}_1 \\ \hat{\mathbf{k}}_2 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{f}}_1 \\ \hat{\mathbf{f}}_2 \end{bmatrix} \right).$$

Convergence of block-prec Krylov defined by preconditioning of Schur complement,

$$\begin{aligned} S &:= \eta I - \delta t \mathcal{L}_2 + \beta^2 (\eta I - \delta t \mathcal{L}_1)^{-1} \\ &\approx \eta I - \delta t \mathcal{L}_2 + \frac{\beta^2}{\eta} I. \end{aligned}$$

# Linear block preconditioning

Requires solution of  $s/2$  systems

$$\begin{bmatrix} \eta I - \delta t \mathcal{L}_1 & \mathbf{0} \\ -\frac{\beta^2}{\phi} I & \frac{\eta^2 + \beta^2}{\eta} I - \delta t \mathcal{L}_2 \end{bmatrix}^{-1} \left( \begin{bmatrix} \eta I - \delta t \mathcal{L}_1 & \phi I \\ -\frac{\beta^2}{\phi} I & \eta I - \delta t \mathcal{L}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{k}}_1 \\ \hat{\mathbf{k}}_2 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{f}}_1 \\ \hat{\mathbf{f}}_2 \end{bmatrix} \right).$$

**Theorem:** Assume  $W(\mathcal{L}_i) \leq 0$  and  $\eta > 0$ :

Stages	2	3		4		5		
	$\lambda_{1,2}^\pm$	$\lambda_1$	$\lambda_{2,3}^\pm$	$\lambda_{1,2}^\pm$	$\lambda_{3,4}^\pm$	$\lambda_1$	$\lambda_{2,3}^\pm$	$\lambda_{4,5}^\pm$
Gauss	1.17	1.00	1.46	1.80	1.05	1.00	2.18	1.14
Radau IIA	1.25	1.00	1.65	2.11	1.06	1.00	2.60	1.16
Lobatto IIIC	1.50	1.00	2.11	2.76	1.07	1.00	3.44	1.19

**Table:** Upper bound on condition number of preconditioned operator for  $\mathcal{L}_1 = \mathcal{L}_2$  ( $\times 2$  for  $\mathcal{L}_1 \neq \mathcal{L}_2$ ).

# Compressible Euler vortex

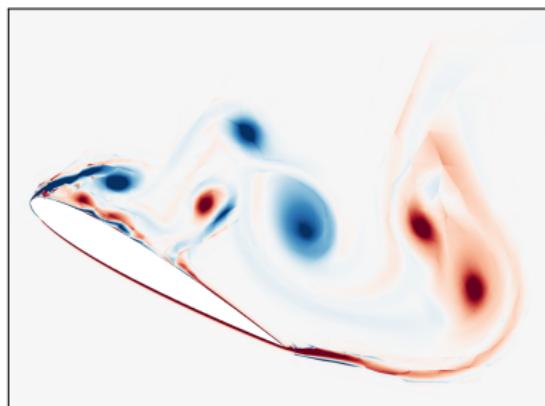
- 4th-order DG discretization in space, Roe numerical fluxes.
- Block-ILU preconditioning for systems ( $\gamma M - \mathcal{L}$ ).

Table: Error and convergence rates for Euler vortex problem

$\Delta t$	Gauss 2		Gauss 4		Gauss 6	
	Error	Rate	Error	Rate	Error	Rate
$2.50 \times 10^{-2}$	$5.89 \times 10^{-3}$	—	$5.29 \times 10^{-4}$	—	$1.65 \times 10^{-5}$	—
$1.25 \times 10^{-2}$	$1.18 \times 10^{-3}$	2.32	$2.75 \times 10^{-5}$	4.26	$2.35 \times 10^{-7}$	6.14
		Radau 3		Radau 5		Radau 7
$2.50 \times 10^{-2}$	$1.19 \times 10^{-3}$	—	$8.48 \times 10^{-5}$	—	$1.92 \times 10^{-6}$	—
$1.25 \times 10^{-2}$	$1.62 \times 10^{-4}$	2.88	$2.78 \times 10^{-6}$	4.92	$1.68 \times 10^{-8}$	6.84

# Compressible Re=40,000 NACA Airfoil

- No-slip BCs at surface of airfoil, farfield BCs at other boundaries.
- Boundary layer at surface of airfoil resolved using layer of anisotropically stretched elements  $\Rightarrow$  highly restrictive CFL.
- 3rd-order DG spatial discretization, time step  $\delta t = 5 \times 10^{-2}$  (several orders of magnitude  $>$  explicit CFL).



# Compressible Re-40,000 NACA Airfoil

- No-slip BCs at surface of airfoil, farfield BCs at other boundaries.
- Boundary layer at surface of airfoil resolved using layer of anisotropically stretched elements  $\Rightarrow$  highly restrictive CFL.
- 3rd-order DG spatial discretization, time step  $\delta t = 5 \times 10^{-2}$  (several orders of magnitude  $>$  explicit CFL).

Order	SDIRK				Gauss				
	1	2	3	4	2	4	6	8	10
Newton its.	5	5	5	5	5	5	8	8	8
Prec. apps.	173	200	359	481	128	244	557	732	830

Table: Nonlinear iterations and preconditioner applications with relative nonlinear tolerance of  $10^{-9}$ .

# Incompressible Navier–Stokes in vorticity formulation

$$\begin{aligned}\partial_t w + \mathbf{v} \cdot \nabla w - \nabla_{\perp} \cdot \mu \nabla_{\perp} w &= 0, \\ -\nabla_{\perp}^2 \phi + w &= 0, \\ \mathbf{v} + \nabla \times \phi \hat{\mathbf{z}} &= \mathbf{0}.\end{aligned}$$

⇒ AIR-AMG for  $w$ , classical AMG for  $\phi$ .

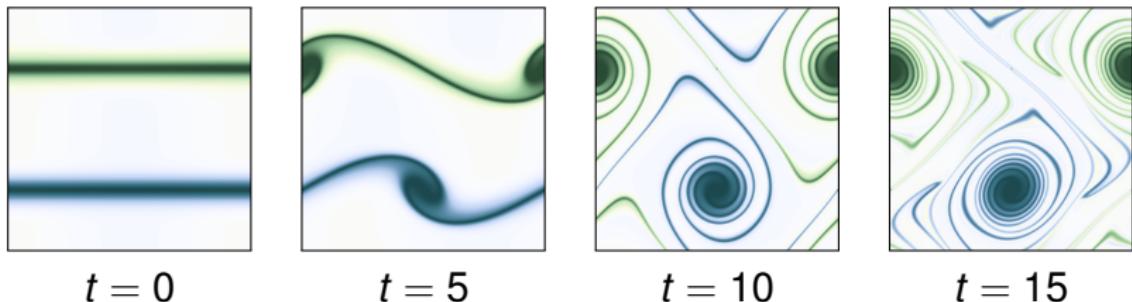


Figure: Time evolution of vorticity for the double shear layer problem.

# Incompressible Navier–Stokes in vorticity formulation

Double shear layer problem, 3rd-order DG elements,  $h = 0.0025$ :

$\delta t$	Gauss 4		Gauss 6		Gauss 8	
	Error	Rate	Error	Rate	Error	Rate
0.4	$1.98 \times 10^{-3}$	—	$1.59 \times 10^{-4}$	—	$4.13 \times 10^{-5}$	—
0.2	$1.30 \times 10^{-4}$	3.93	$9.08 \times 10^{-7}$	7.45	$1.04 \times 10^{-8}$	11.95
Radau 5		Radau 7		Radau 9		
0.4	$2.37 \times 10^{-4}$	—	$3.96 \times 10^{-6}$	—	$6.21 \times 10^{-8}$	—
0.2	$8.35 \times 10^{-6}$	4.83	$3.54 \times 10^{-8}$	6.81	$1.42 \times 10^{-10}$	8.77

**Table:**  $Re=10$  error and convergence w.r.t. ref solution from 6th-order explicit RK with  $\delta t = 10^{-4}$  (roughly CFL limit).

# Incompressible Navier–Stokes in vorticity formulation

Double shear layer problem, 3rd-order DG elements,  $h = 0.0025$ :

Order	SDIRK				Gauss				
	1	2	3	4	2	4	6	8	10
Prec. apps.	127	141	322	365	78	161	218	287	333

Table: Preconditioner applications,  $\text{Re} = 10$ .

Order	SDIRK				Gauss				
	1	2	3	4	2	4	6	8	10
Prec. apps.	41	72	113	177	37	75	118	163	194

Table: Preconditioner applications,  $\text{Re} = 25,000$ .

# Table of Contents

Fast solvers for fully implicit integration

Numerical results

Fully implicit-explicit integration

Polynomial integrators

Implicit-explicit Radau

Numerical results

# Additive integrators

$y' = f(t, y)$ :

$$\implies y' = f^{\{1\}}(t, y) + f^{\{2\}}(t, y), \quad y(t_0) = y_0.$$

# Additive integrators

$y' = f(t, y)$ :

$$\implies y' = f^{\{1\}}(t, y) + f^{\{2\}}(t, y), \quad y(t_0) = y_0.$$

Discrete integral form:

$$y(t_{n+1}) = y(t_n) + \underbrace{\int_{t_n}^{t_{n+1}} f^{\{1\}}(t, y(t)) dt}_{\text{treat implicitly}} + \underbrace{\int_{t_n}^{t_{n+1}} f^{\{2\}}(t, y(t)) dt}_{\text{treat explicitly}}.$$

# Polynomial integrators

Many numerical integrators are based on polynomial interpolation (Adams-Basforth, Adams-Moulton, BDF, etc.).

# Polynomial integrators

Many numerical integrators are based on polynomial interpolation (Adams-Basforth, Adams-Moulton, BDF, etc.).

⇒ Construct continuous polynomials in time that fit solution or derivative values at nodes  $\{z_j\} \subset [-1, 1]$ .<sup>1</sup>

$$\text{input (solutions): } y_j^{[n]} \approx y(t_n + rz_j)$$

$$\text{output (solutions): } y_j^{[n+1]} \approx y(t_n + rz_j + h)$$

$$\text{input derivatives: } f_j^{[n]} := f(t_n + rz_j, y_j^{[n]}) \approx y'(t_n + rz_j)$$

$$\text{output derivatives: } f_j^{[n+1]} := f(t_n + rz_j + h, y_j^{[n+1]}) \approx y'(t_n + rz_j + h).$$

---

<sup>1</sup>In practice, better to shift nodes to  $[0, 2]$ ; does not effect method coefficients.

# Polynomial block methods

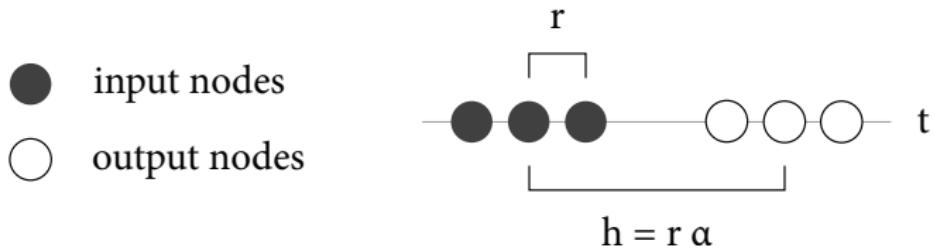
Construct degree- $g$  ODE polynomial to approximate Taylor series of solution in local coordinates  $\tau$ , where  $t(\tau) = r\tau + t_n$ :

$$y(t(\tau)) \approx p(\tau; b) := \sum_{j=0}^g \frac{a_j(b)(\tau - b)^j}{j!}, \quad \text{where } a_j(b) \approx \left. \frac{d^j}{d\tau^j} y(t(\tau)) \right|_{\tau=b}$$

# Polynomial block methods

Construct degree- $g$  ODE polynomial to approximate Taylor series of solution in local coordinates  $\tau$ , where  $t(\tau) = r\tau + t_n$ :

$$y(t(\tau)) \approx p(\tau; b) := \sum_{j=0}^g \frac{a_j(b)(\tau - b)^j}{j!}, \quad \text{where } a_j(b) \approx \left. \frac{d^j}{d\tau^j} y(t(\tau)) \right|_{\tau=b}$$



**Figure:** Input and output nodes for polynomial integrator with  $q = 3$  and nodes  $z_j = \{-1, 0, 1\}$ ;  $r := \text{node radius}$ ,  $\alpha := \text{extrapolation factor}$ .

## Adams ODE polynomials

Two Lagrange interpolating polynomials:

- $L_y(\tau) \approx y(t(\tau))$ : interpolates at least one solution value.
- $L_f(\tau) \approx ry'(t(\tau)) = rf(t(\tau))$ : polynomial of degree  $g - 1$ ,  
interpolates  $g$  derivative values.

$$a_0(b) = L_y(b), \quad \text{and} \quad a_j(b) = \frac{d^{j-1} L_f}{d\tau^{j-1}}(b), \quad j = 1, \dots, g.$$

# Adams ODE polynomials

Two Lagrange interpolating polynomials:

- $L_y(\tau) \approx y(t(\tau))$ : interpolates at least one solution value.
- $L_f(\tau) \approx ry'(t(\tau)) = rf(t(\tau))$ : polynomial of degree  $g - 1$ ,  
interpolates  $g$  derivative values.

$$a_0(b) = L_y(b), \quad \text{and} \quad a_j(b) = \frac{d^{j-1} L_f}{d\tau^{j-1}}(b), \quad j = 1, \dots, g.$$

Noting  $p'(\tau; b) = L_f(\tau)$ , the Adams ODE polynomial can be written

$$p(\tau; b) = L_y(b) + \int_b^\tau L_f(\xi) d\xi \quad \leftarrow \quad \approx \int_b^\tau f(t(\xi), y) d\xi$$

## Example: Adams-methods

Classical linear multistep methods: can think of as block method, where first  $q - 1$  output nodes = last  $q - 1$  input nodes.

## Example: Adams-methods

Classical linear multistep methods: can think of as block method, where first  $q - 1$  output nodes = last  $q - 1$  input nodes.

**Adams-Basforth (explicit):** For output nodes  $\{\tau_j\}_{j=1}^q$ , construct  $L_f(\tau)$  to satisfy

$$L_f(\tau_j) = f(t(\tau_j), y_j), \quad \text{for } j = 1, \dots, q - 1.$$

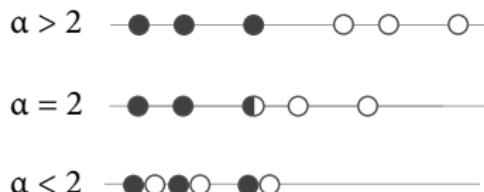
**Adams-Moulton (implicit):** For output nodes  $\{\tau_j\}_{j=1}^q$ , construct  $L_f(\tau)$  to satisfy

$$L_f(\tau_j) = f(t(\tau_j), y_j), \quad \text{for } j = 1, \dots, q.$$

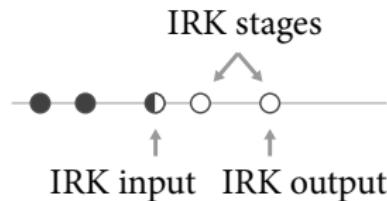
## Example: RadauIA

Let  $x_j := j$ th zero of  $\frac{d^{q-2}}{dx^{q-2}}(x^{q-2}(x-1)^{q-1})$ . Choose

$$\{z_j\} = \left\{ -1, \underbrace{2x_1 - 1, \dots, 2x_{q-1} - 1}_{\text{Radau nodes}} \right\}.$$



(a) PBM input and output nodes



(b) Equivalent RK inputs, outputs, stages

**Figure:** Radau input nodes ● and output nodes ○ with  $q = 3$   
( $\{z_j\} = \{-1, -1/3, 1\}$ ). Gray lines show time axis which flows to right.

# Fully implicit-explicit (RadauIIA-Bashforth)

Recall discrete integral form of additive method:

$$y(t_{n+1}) = y(t_n) + \underbrace{\int_{t_n}^{t_{n+1}} f^{\{1\}}(t, y(t)) dt}_{\text{treat implicitly}} + \underbrace{\int_{t_n}^{t_{n+1}} f^{\{2\}}(t, y(t)) dt}_{\text{treat explicitly}}.$$

# Fully implicit-explicit (RadauIA-Bashforth)

*Partitioned* Adams ODE polynomial:

$$p(\tau; b) = L_y(b) + \underbrace{\int_b^\tau L_f^{\{1\}}(\xi) d\xi}_{\text{implicit approx.}} + \underbrace{\int_b^\tau L_f^{\{2\}}(\xi) d\xi}_{\text{explicit approx.}}$$

# Fully implicit-explicit (RadauIIA-Bashforth)

*Partitioned* Adams ODE polynomial:

$$p(\tau; b) = L_y(b) + \underbrace{\int_b^\tau L_f^{\{1\}}(\xi) d\xi}_{\text{implicit approx.}} + \underbrace{\int_b^\tau L_f^{\{2\}}(\xi) d\xi}_{\text{explicit approx.}}$$

- $L_y(\tau) :=$  degree  $q - 1$  polynomial that interpolates all  $q$  inputs,  
 $L_y(z_j) = y_j^{[n]}, j = 1, \dots, q.$

# Fully implicit-explicit (RadauIIA-Bashforth)

*Partitioned* Adams ODE polynomial:

$$p(\tau; b) = L_y(b) + \underbrace{\int_b^\tau L_f^{\{1\}}(\xi) d\xi}_{\text{implicit approx.}} + \underbrace{\int_b^\tau L_f^{\{2\}}(\xi) d\xi}_{\text{explicit approx.}}$$

- $L_y(\tau) :=$  degree  $q - 1$  polynomial that interpolates all  $q$  inputs,  
 $L_y(z_j) = y_j^{[n]}, j = 1, \dots, q.$
- $L_f^{\{1\}}(\tau) :=$  degree  $q - 2$  polynomial that interpolates last  $q - 1$  output derivatives,  
 $L_f^{\{1\}}(z_j + \alpha) = rf_j^{\{1\}[n+1]}, j = 2, \dots, q$

# Fully implicit-explicit (RadauIA-Bashforth)

*Partitioned* Adams ODE polynomial:

$$p(\tau; b) = L_y(b) + \underbrace{\int_b^\tau L_f^{\{1\}}(\xi) d\xi}_{\text{implicit approx.}} + \underbrace{\int_b^\tau L_f^{\{2\}}(\xi) d\xi}_{\text{explicit approx.}}$$

- $L_y(\tau) :=$  degree  $q - 1$  polynomial that interpolates all  $q$  inputs,  
 $L_y(z_j) = y_j^{[n]}, j = 1, \dots, q.$
- $L_f^{\{1\}}(\tau) :=$  degree  $q - 2$  polynomial that interpolates last  $q - 1$  output derivatives,  
 $L_f^{\{1\}}(z_j + \alpha) = rf_j^{\{1\}[n+1]}, j = 2, \dots, q$
- $L_f^{\{2\}}(\tau) :=$  degree  $q - 1$  polynomial that interpolates input derivative components,  
 $L_f^{\{2\}}(z_j) = rf_j^{\{2\}[n]}, j = (1 \text{ or } 2), \dots, q.$

# Fully implicit-explicit (RadauIIA-Bashforth)

Partitioned Adams ODE polynomial:

$$p(\tau; b) = L_y(b) + \underbrace{\int_b^\tau L_f^{\{1\}}(\xi) d\xi}_{\text{implicit approx.}} + \underbrace{\int_b^\tau L_f^{\{2\}}(\xi) d\xi}_{\text{explicit approx.}}$$

General coefficient form:

$$\mathbf{y}^{[n+1]} = \mathbf{A}\mathbf{y}^{[n]} + r\mathbf{B}^{\{1\}}\mathbf{f}^{\{1\}}[n+1] + r\mathbf{B}^{\{2\}}\mathbf{f}^{\{2\}}[n]$$

E.g., for  $q = 3$ :

$$\begin{bmatrix} y_1^{n+1} \\ y_2^{n+1} \\ y_3^{n+1} \end{bmatrix} = \begin{bmatrix} y_1^n \\ y_2^n \\ y_3^n \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{5}{6} & -\frac{1}{6} \\ 0 & \frac{3}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} f_1^{\{1\},n+1} \\ f_2^{\{1\},n+1} \\ f_3^{\{1\},n+1} \end{bmatrix} + r \begin{bmatrix} 0 & 0 & 0 \\ \frac{8}{27} & -\frac{11}{18} & \frac{53}{54} \\ 4 & -\frac{15}{2} & \frac{11}{2} \end{bmatrix} \begin{bmatrix} f_1^{\{2\},n} \\ f_2^{\{2\},n} \\ f_3^{\{2\},n} \end{bmatrix}$$

⇒ same solvers as IRK with modified right-hand side!

# Iterators

Order-of-accuracy = minimum order of implicit and explicit component.

- *Implicit*: Radau IIA with  $q - 1$  stages  $\rightarrow$  order  $2q - 3$ .
- *Explicit*: order  $q - 1$  or  $q$ , depending if use first node.

# Iterators

Order-of-accuracy = minimum order of implicit and explicit component.

- *Implicit*: Radau IIA with  $q - 1$  stages  $\rightarrow$  order  $2q - 3$ .
- *Explicit*: order  $q - 1$  or  $q$ , depending if use first node.

Recall  $h = r\alpha$ . Set  $\alpha = 0 \mapsto$  special method called *iterator*.

- Iterator updates solution *in place*.
- Each application of iterator (formally) increases order by one; converges to solution of fully implicit integrator.
- Iterator also increases stability region.

# Iterators

Order-of-accuracy = minimum order of implicit and explicit component.

- *Implicit*: Radau IIA with  $q - 1$  stages  $\rightarrow$  order  $2q - 3$ .
- *Explicit*: order  $q - 1$  or  $q$ , depending if use first node.

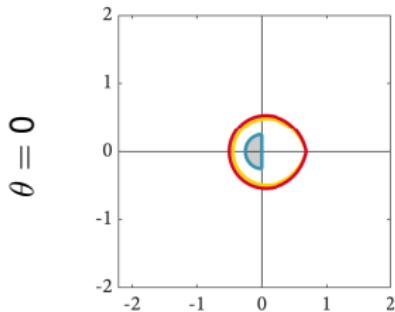
Recall  $h = r\alpha$ . Set  $\alpha = 0 \mapsto$  special method called *iterator*.

- Iterator updates solution *in place*.
- Each application of iterator (formally) increases order by one; converges to solution of fully implicit integrator.
- Iterator also increases stability region.

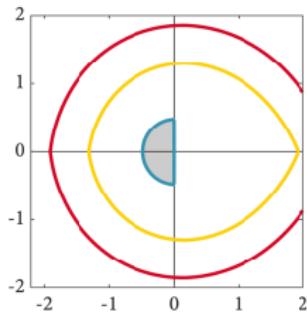
**IMEX-Radau( $q, \kappa$ ) and IMEX-Radau\* $(q, \kappa)$**

# Stability and iterator, IMEX-Radau\*( $q = 4, \kappa$ )

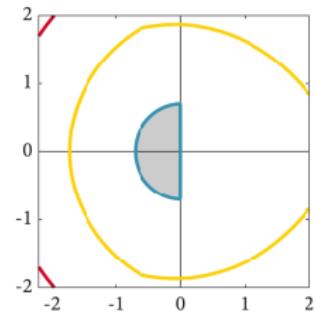
$\kappa = 0$



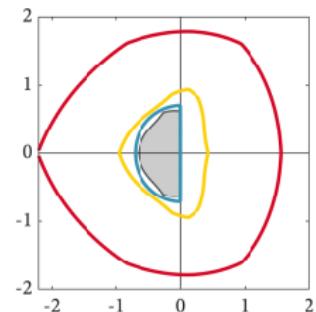
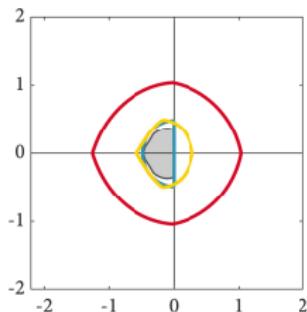
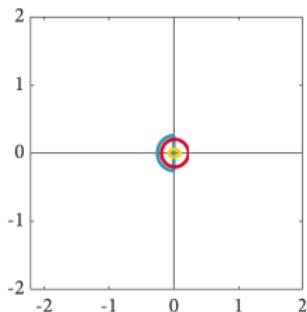
$\kappa = 1$



$\kappa = 2$



$\|\theta\|$



■  $|r| = 0$  ■  $|r| = 3$  ■  $|r| = 6$

## DG advection-diffusion

IP-DG discretization of time-dependent advection-diffusion:

$$u_t + \nabla \cdot ([1, 1] \cdot u - \varepsilon \nabla u) = f,$$

$\varepsilon$  = diffusion coefficient,  $f(x, y, t)$  s.t. solution given by

$$u_*(x, y, t) = \sin(2\pi x(1 - y)(1 + 2t)) \sin(2\pi y(1 - x)(1 + 2t)).$$

## DG advection-diffusion

IP-DG discretization of time-dependent advection-diffusion:

$$u_t + \nabla \cdot ([1, 1] \cdot u - \varepsilon \nabla u) = f,$$

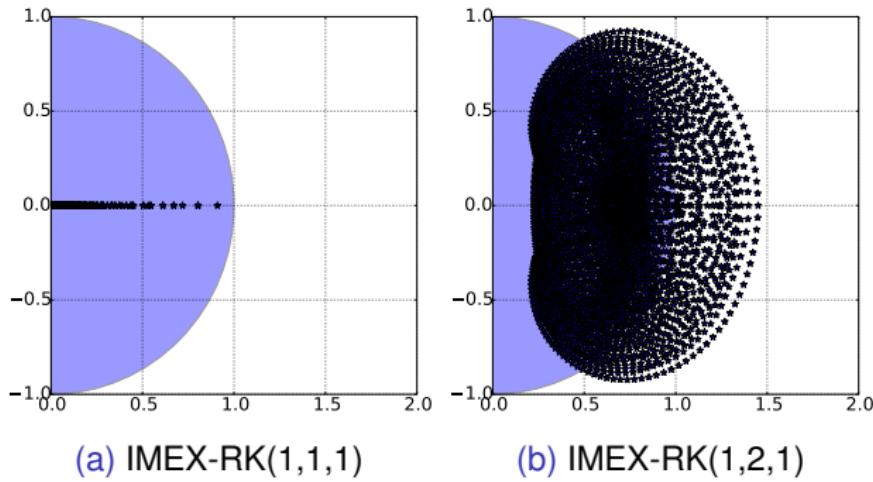
$\varepsilon$  = diffusion coefficient,  $f(x, y, t)$  s.t. solution given by

$$u_*(x, y, t) = \sin(2\pi x(1 - y)(1 + 2t)) \sin(2\pi y(1 - x)(1 + 2t)).$$

Compare

- RK(1,1,1), RK(2,2,2), RK(4,4,3) from Ascher et al.
- ESDIRK ARK3(2)4L[2]SA ( $\mapsto$  ARK(4,3)) from Kennedy/Carpenter

# RK instability



**Figure:** Eigenvalues of propagation operator for two first-order IMEX-RK schemes from Ascher et al., with time-step given by advective stability limit. Eigenvalues must be inside shaded region for stable time propagation.

# IMEX-Radau\*

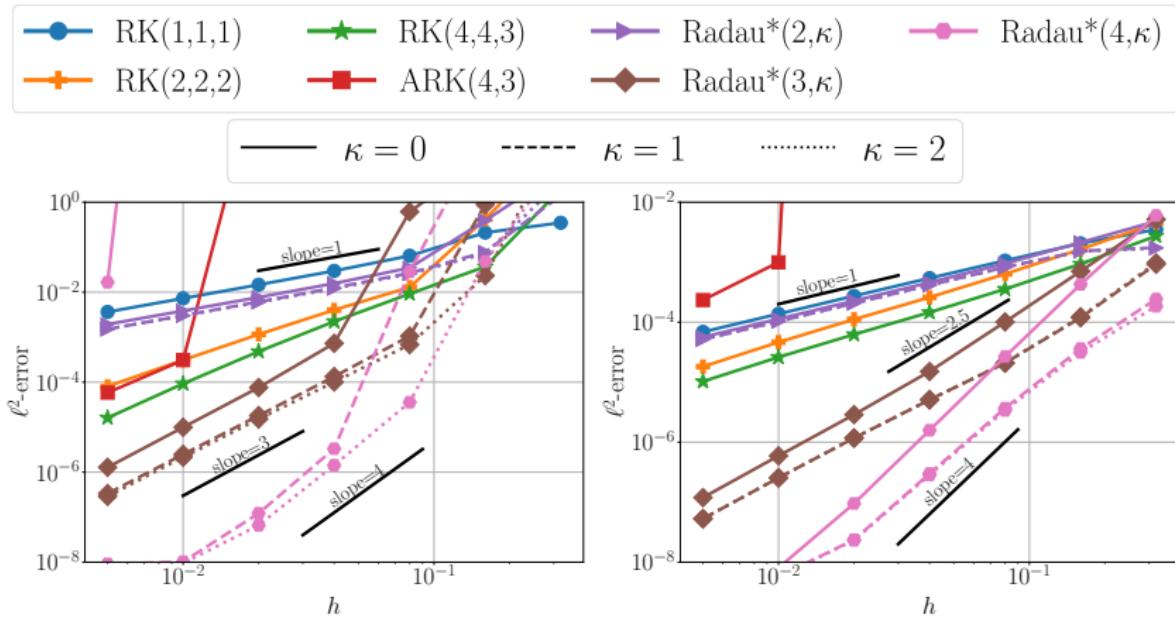
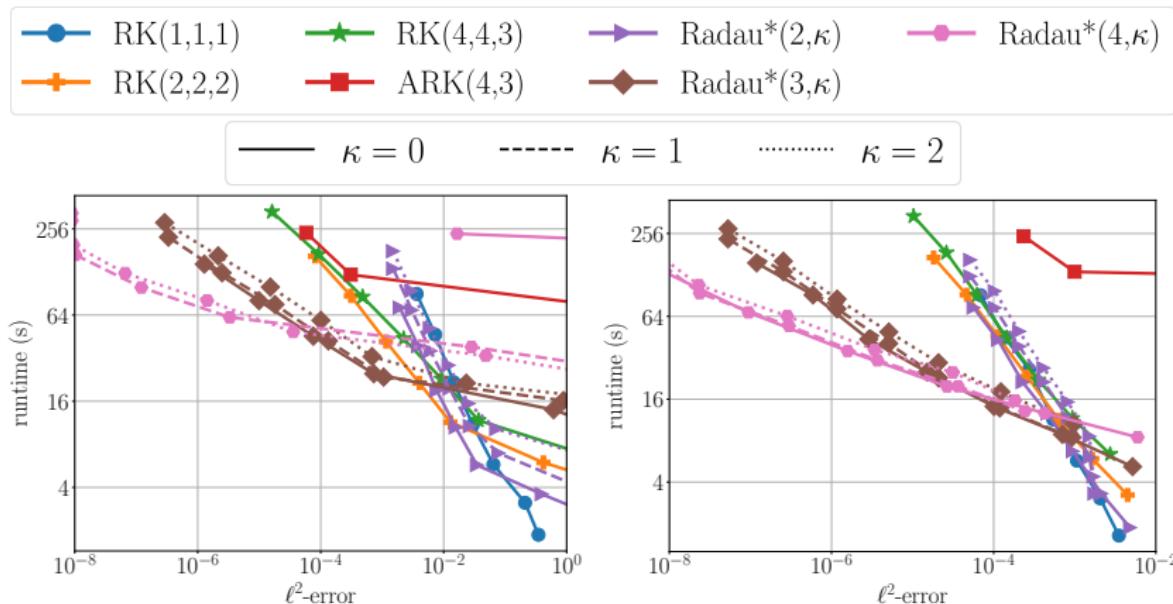


Figure:  $\ell^2$ -accuracy as a function of  $h$ .

# IMEX-Radau\*



(a)  $\epsilon = 0.01$

(b)  $\epsilon = 10$

Figure: Runtime as a function of  $\ell^2$ -accuracy.

## Ongoing and todos

- Preconditioners and IMEX-Radau for index-2 DAEs (Navier Stokes).
- A finite-element interpretation of the IMEX methods.
- Other new classes of integrators, e.g., fully-implicit(-explicit) BDF, multirate integrators, etc.

# Thank you!

## Papers:

T. Buvoli and B. S. Southworth. *Additive Polynomial Methods, Part I: Framework and Fully-Implicit Block Methods*.

B. S. Southworth et al. *Fast parallel solution of fully implicit Runge-Kutta and discontinuous Galerkin in time for numerical PDEs, Part I: the linear setting*.

B. S. Southworth, O. Krzysik, and W. Pazner. *Fast parallel solution of fully implicit Runge-Kutta and discontinuous Galerkin in time for numerical PDEs, Part II: nonlinearities and DAEs*.